In the claims:

Please substitute the following full listing of claim for the claims as originally filed or most recently amended.

1. (Original) A method of operating a digital system controlled by operation codes and operable in a plurality of operational modes, said method comprising steps of

processing an application program to insert execution bits in operational codes preceding instructions which are not used in ones of said plurality of modes,

evaluating each operational code prior to decoding of said each operational code,

skipping an operational code responsive to detecting a particular state of an execution bit in a preceding instruction, and

decoding remaining operational codes.

2. (Original) A method as recited in claim 1, wherein said processing is performed in response to a criterion representing one of an operating mode, a function and a peripheral device connected to said digital system.

3. (Original) A method as recited in claim 1, wherein said skipping step skips a single operational code following said preceding instruction.

4. (Original) A method as recited in claim 1, wherein said skipping step skips all operations between said preceding instruction and another instruction having an execution bit having said particular state.

5. (Original) A method as recited in claim 4, wherein said skipping step is performed by toggling a bit in a register upon detection of an activation bit in said particular state and iteratively comparing execution bits of instructions with said bit in said register.

6. (Original) A method as recited in claim 1, wherein a plurality of execution bits are provided in each instruction by said processing step and said skipping step is performed in accordance with a number represented by said plurality of execution bits.

7. (Original) A method as recited in claim 2, wherein said skipping step skips a single operational code following said preceding instruction.

8. (Original) A method as recited in claim 2, wherein said skipping step skips all operations between said preceding instruction and another instruction having an execution bit having said particular state.

9. (Original) A method as recited in claim 8, wherein said skipping step is performed by toggling a bit in a register upon detection of an activation bit in said particular state and iteratively comparing execution bits of instructions with said bit in said register.

10. (Original) A method as recited in claim 2, wherein a plurality of execution bits are provided in each instruction by said processing step and said skipping step is performed in accordance with a number represented by said plurality of execution bits.

11. (Currently Amended) A processor comprising

means for processing and storing a sequence of instructions to be available for execution, each said instruction including an execution bit, and

means for bypassing said processing of ~~an~~ a following instruction of said sequence of instructions based on a particular state of an execution bit in a current instruction.

12. (Original) A processor as recited in claim 11, further including

means for indicating an operating state of said processor for setting a criterion for processing instructions of an application program for providing execution bits of respective states in said instructions.

13. (Original) A processor as recited in claim 11, wherein said instruction of said sequence bypassed by said means for bypassing follows said instruction having said execution bit of said particular state.

14. (Original) A processor as recited in claim 11, wherein said means for bypassing includes means for selectively bypassing a plurality of said instructions.

15. (Original) A processor as recited in claim 14, further including

means for bypassing instructions between instructions having execution bits of said particular state.

16. (Original) A processor as recited in claim 15, further including

means for toggling a comparison bit upon detection of an execution bit of said particular state, and

means for comparing execution bits of respective instructions with said comparison bit.

17. (Original) A processor as recited in claim 14. wherein instructions of said plurality of instructions include a plurality of execution bits and wherein said bypassing means includes means for bypassing a plurality of instructions corresponding to said plurality of execution bits.